

ΕΡΓΑΣΙΑ 1^η

Ημερομηνία Παράδοσης: 25-11-2020

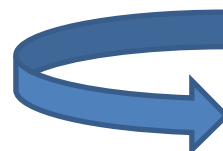
Φοιτητής: ΓΙΑΝΝΑΚΟΣΙΑΝ ΑΝΕΣΤΗΣ

Τμήμα: ΠΛΣ50-ΗΛΕ45

Υπεύθυνος Καθηγητής: Κος ΜΑΥΡΟΜΜΑΤΗΣ ΓΕΩΡΓΙΟΣ

Πρόλογος

Πριν ξεκινήσει η επεξήγηση των επιμέρους θεμάτων της Εργασίας 1, να αναφέρω πως τα προγράμματα δημιουργήθηκαν στο **IntelliJ IDEA** της εταιρείας JETBRAINS, με JDK version 15, αλλά δοκιμάστηκαν και στο **Bluej** για την εύρυθμη λειτουργία τους. Επίσης θα ήθελα να κάνω γνωστό ότι παρακολουθώ παράλληλα, ένα Java course μέσω του Udemy.com, για να μπορώ να έχω πρόσβαση σε ακόμα περισσότερες γνώσεις στο αντικείμενο και να εξασκούμε περισσότερο, κάνοντας όσο το δυνατόν περισσότερες ασκήσεις. Αυτό γίνεται μέσω βιντεοσκοπημένων μαθημάτων και πληθώρα coding exercises, για κάθε κεφάλαιο που εξετάζουμε, για τη γλώσσα προγραμματισμού Java.



Θέμα 1: Ψηφοφορία για την επιλογή νησιού για διακοπές

Ψηφοφορία του κατάλληλου νησιού για διακοπές. Στην ψηφοφορία συμμετέχουν 20 άτομα και κάθε άτομο ψηφίζει μόνο μια φορά. Θεωρούμε ότι τα υποψήφια νησιά που καλούνται τα άτομα αυτά να ψηφίσουν για διακοπές είναι η Μύκονος, η Σαντορίνη και η Ρόδος...

Να γράψετε πρόγραμμα Java που θα αποτελείται από μια κλάση με το όνομα Holiday και:

- Θα ζητάει από το χρήστη να ψηφίσει το νησί που θεωρεί τον καλύτερο προορισμό για διακοπές, εισάγοντας από το πληκτρολόγιο 1 για την Μύκονο, 2 για την Σαντορίνη και 3 για την Ρόδο. Στην συνέχεια, θα τυπώνει στην οθόνη “Ψηφίσατε ως κατάλληλο νησί ...”, όπου στην θέση των αποσιωπητικών (...) θα εμφανίζεται το όνομα του νησιού που ψηφίστηκε ή το μήνυμα “Η ψήφος σας δεν ήταν έγκυρη!” σε περίπτωση που δώσει κάτι διαφορετικό από 1, 2, 3,

Αφού ψηφίσουν όλοι, το πρόγραμμα:

- θα υπολογίζει πόσες ψήφους πήρε το κάθε νησί και το ποσοστό (%) επί του συνόλου των έγκυρων ψήφων,
- θα προβάλλει το νησί με τις λιγότερες ψήφους και τον καλύτερο προορισμό για διακοπές δηλαδή το νησί με τις περισσότερες ψήφους.

➤ Υλοποίηση Holiday.java

Το πρόγραμμα αποτελείται από μία κλάση μέσα στην οποία βρίσκεται και η main() μέθοδος. Κάνοντας import την κλάση Scanner επιτρέπουμε στο πρόγραμμα να δέχεται καταχωρήσεις από τον user.

A) Μεταβλητές Κλάσης

Οι 4 στατικές μεταβλητές που δηλώθηκαν είναι οι εξής:

- *private static final Scanner **input** = new Scanner (System.in).*
- *private static final int[] **validVotesArray** = {0,0,0},* η οποία είναι ένας μονοδιάστατος πίνακας 3 θέσεων για την αποθήκευση των έγκυρων ψήφων του κάθε νησιού που συμμετέχει στην ψηφοφορία.
- *private static int **totalValidVotes**,* η οποία θα αποθηκεύσει τις συνολικές έγκυρες ψήφους κατά την κύρια διαδικασία της ψηφοφορίας για να μας βοηθήσει να υπολογίσουμε το ποσοστό των έγκυρων ψήφων του κάθε νησιού.
- *private static final String **answers[]**,* η οποία είναι ένας μονοδιάστατος πίνακας 3 θέσεων όπου σε κάθε θέση αντιστοιχεί και ένα υποψήφιο νησί, με την σειρά που δηλώθηκαν.

B) Ανάλυση μεθόδων

Η κλάση αποτελείται από 6 μεθόδους, ως εξής:

- *public static void **main**(String[] args).* Η main() καλεί την **runHolidayPollProcess()** και εκτελείται το πρόγραμμα.
- *public static void **runHolidayPollProcess()**.* Η συγκεκριμένη μέθοδος καλεί όλες τις υπόλοιπες μεθόδους της κλάσης που εκτελούν επιμέρους λειτουργίες μέσα σε μία for loop. Η for loop εκτελείται 20 φορές, μέχρι τη στιγμή δηλαδή που μαζί με έγκυρες και άκυρες ψήφους θα έχουμε 20 μοναδικές ψήφους.
- *public static String **interactWithUser()**.* Η συγκεκριμένη μέθοδος προβάλλει στην οθόνη την βασική ερώτηση της ψηφοφορίας σύμφωνα με την εκφώνηση της άσκησης. Επίσης χρησιμοποιεί μία for loop για να εμφανίσει το αντίστοιχο ζευγάρι ψηφίου (i+1) - υποψήφιου νησιού, με χρήση του πίνακα

answers[i]. Διαβάζει την καταχώρηση του user στο terminal και την επιστρέφει input.nextLine(). Να προσθέσω εδώ ότι αποφάσισα να βάλω το .nextLine(), για να γίνεται δεκτή η όποια κενή καταχώριση, προηγείται της χρήσης του πλήκτρου Enter για επιβεβαίωση και να εκλαμβάνεται ως άκυρη ψήφος.

- **public static void validateVotes(String vote).** Η μέθοδος αυτή είναι υπεύθυνη για την επικύρωση των καταχωρήσεων (votes). Μέσα σε μία switch statement εξετάζεται εάν το input του χρήστη είναι "1", "2" ή "3" και ενημερώνει τον χρήστη σε ποιο νησί αντιστοιχεί η ψήφος. Παράλληλα έχουμε την προσαύξηση της μεταβλητής **totalValidVotes**, καθώς και την προσαύξηση κατά ένα στην αντίστοιχη θέση στον πίνακα **validVotesArray[i]**. Ενδεικτικά στην θέση validVotesArray[0] αποθηκεύονται οι έγκυρες ψήφοι για την Μύκονο, στην validVotesArray[1] για την Σαντορίνη και στην validVotesArray[2] για την Ρόδο. Αν ο χρήστης πληκτρολογήσει οποιοδήποτε άλλο χαρακτήρα στο terminal είτε κενή καταχώρηση θα εμφανίζεται μήνυμα άκυρης ψήφου.
- **public static void printWinner().** Δηλώθηκαν οι τοπικές μεταβλητές maxValidVotes και maxPosition, τύπου integer. Μέσα σε μία for loop και για 3 φορές, όσο είναι το μέγεθος .length του πίνακα validVotesArray[i], κάνοντας σύγκριση με τη βοήθεια μίας if statement υπολογίζει ποιο νησί πήρε τις περισσότερες ψήφους και αποθηκεύει τις ψήφους αυτές στην μεταβλητή maxValidVotes. Επομένως το νησί που θεωρείται υπερψηφισμένο θα ορισθεί στη θέση maxPosition = i, και θα αναζητηθεί στον πίνακα answers[maxPosition]. Τέλος η μέθοδος υπολογίζει το ποσοστό επί του συνόλου έγκυρων ψήφων (%), κάνει στρογγυλοποίηση στο δεύτερο δεκαδικό ψηφίο με τη χρήση της Math.round και επιστρέφει τα αποτελέσματα.
- **public static void printLoser().** Δηλώθηκαν οι τοπικές μεταβλητές minValidVotes και minPosition, τύπου integer. Ακολουθώντας παρόμοια διαδικασία με την printWinner(), υπολογίζει ποιο νησί πήρε τις λιγότερες ψήφους και αποθηκεύει τις ψήφους αυτές στην μεταβλητή minValidVotes. Επομένως το νησί που θεωρείται λιγότερο ψηφισμένο θα ορισθεί αυτό στη θέση minPosition = i, και θα αναζητηθεί στον πίνακα answers[minPosition]. Τέλος η μέθοδος υπολογίζει το ποσοστό επί του συνόλου έγκυρων ψήφων (%), κάνει στρογγυλοποίηση στο δεύτερο δεκαδικό ψηφίο με τη χρήση της Math.round και επιστρέφει τα αποτελέσματα.

Αξίζει να σημειωθεί τέλος ότι δεν έχει δημιουργηθεί κάποια μέθοδος που θα αντιμετωπίσει την περίπτωση ισοψηφίας καθώς δεν ήταν ζητούμενο της άσκησης. Σε περίπτωση ισοψηφίας δύο νησιών ή ολικής ισοψηφίας, θα εμφανιστεί ως νησί με τις περισσότερες ψήφους αυτό που βρίσκεται πιο κοντά/ή στη θέση answers[0]. Αντίστοιχη λογική και για το νησί με τις λιγότερες ψήφους, εμφανίζοντας αυτό που βρίσκεται πιο κοντά στη θέση answers[2].

Θέμα 2: Έλεγχος ΑΜΚΑ

Ο αριθμός Κοινωνικής Ασφάλισης (Α.Μ.Κ.Α) αποτελείται από 11 αριθμητικά ψηφία. Ειδικότερα, ο εντεκαψήφιος αριθμός που τον συνθέτει συγκροτείται από:

(α) το πρώτο τμήμα που είναι εξαψήφιο και δηλώνει την ημερομηνία γεννήσεως (ΗΗ/ΜΜ/ΕΕ) του συγκεκριμένου υπόχρεου,

(β) το δεύτερο τμήμα που είναι τετραψήφιο και δηλώνει τον αύξοντα αριθμό καταχωρίσεως στο οικείο Εθνικό Μητρώο του υπόχρεου (για τους άνδρες ο αριθμός αυτός είναι περιττός, ενώ για τις γυναίκες άρτιος).

(γ) το τρίτο τμήμα που είναι μονοψήφιο και αποτελεί χαρακτήρα ελέγχου που δίδεται από τη μηχανογράφηση.

Να γράψετε πρόγραμμα Java που θα αποτελείται από μία κλάση με όνομα ΑΜΚΑ και:

- θα διαβάσει από το πληκτρολόγιο ένα αλφαριθμητικό,
- θα ελέγχει ότι το μήκος του αλφαριθμητικού αντιστοιχεί σε ΑΜΚΑ και αν όχι θα ενημερώνει το χρήστη με κατάλληλο μήνυμα και θα επαναλαμβάνει το βήμα της ανάγνωσης του αλφαριθμητικού από το πληκτρολόγιο,
- κάνοντας χρήση ανακύκλωσης, θα απομονώνει κάθε χαρακτήρα του αλφαριθμητικού με τη μέθοδο `charAt`, θα ελέγχει αν ο χαρακτήρας αυτός είναι αριθμητικό ψηφίο με τη μέθοδο `isDigit` (και αν όχι θα ενημερώνει τον χρήστη με κατάλληλο μήνυμα και θα επαναλαμβάνει το βήμα της ανάγνωσης του αλφαριθμητικού από το πληκτρολόγιο) και το αποτέλεσμα θα το αποθηκεύει σε κατάλληλο πίνακα,
- θα υπολογίζει την ηλικία του ατόμου βάσει της διαφοράς του έτους γέννησης με το σημερινό έτος. Θεωρείται ότι όλα τα άτομα έχουν γεννηθεί μετά το 1920,
- θα ελέγχει αν το δεύτερο μέρος του ΑΜΚΑ είναι άρτιος ή περιττός και θα προσδιορίζει το φύλο,
- εκτυπώνει την ηλικία και το φύλο και ρωτάει τον χρήστη αν θέλει να επαναλάβει τη διαδικασία ή θέλει να τερματίσει το πρόγραμμα

➤ Υλοποίηση ΑΜΚΑ.java

Το πρόγραμμα αποτελείται από μία κλάση μέσα στην οποία βρίσκεται και η `main()` μέθοδος. Κάνοντας `import` την κλάση `Scanner` επιτρέπουμε στο πρόγραμμα να δέχεται καταχωρήσεις από τον `user`.

A) Μεταβλητές Κλάσης

Οι 2 στατικές μεταβλητές που δηλώθηκαν είναι οι εξής:

- `private static final int amkaAcceptedLength = 11`, το αποδεκτό δηλαδή μήκος του αριθμού Α.Μ.Κ.Α, που θα δίνεται στο terminal.
- `private static final character[] arrayAmkaDigits = [11]`, η οποία είναι ένας μονοδιάστατος πίνακας 11 θέσεων για την αποθήκευση ψηφίων του αριθμού Α.Μ.Κ.Α.

B) Ανάλυση μεθόδων

Η κλάση αποτελείται 05 μεθόδους, ως εξής:

- `public static void main(String[] args)`. Η `main()` καλεί την μέθοδο `runAmkaProcess()` η οποία εκτελείται εντός μίας `do_while` με `statement` την μέθοδο `restartOrExit()`, έως ότου δηλαδή ο χρήστης ζητήσει τον τερματισμό του προγράμματος.
- `public static String interactWithUser()`. Η συγκεκριμένη μέθοδος προβάλλει στην οθόνη την βασική ερώτηση του προγράμματος, σύμφωνα με την εκφώνηση της άσκησης και επιστρέφει το αλφαριθμητικό input του `user`.
- `public static boolean invalidLength(String amka)`. Επιστρέφει `true` όταν η αλφαριθμητική καταχώρηση έχει μήκος διαφορετικό των 11 χαρακτήρων. (`amka.length() != amkaAcceptedLength`)

- *public static boolean **invalidCharacters**(String amka).* Διαχωρίζει την αλφαριθμητική καταχώρηση και εξετάζει ένα χαρακτήρα την φορά μέσα σε μία for loop με την isDigit. Η for loop εκτελείται 11 φορές όσοι και οι χαρακτήρες του amka. Κατά την διαδικασία αυτή και με την χρήση της charAt(i), εφόσον στην καταχώρηση δόθηκαν τα 11 ψηφία, τα τοποθετεί στον πίνακα **arrayAmkaDigits[]** με τη σειρά που πληκτρολογήθηκαν.
- *public static String **validateUserInput**(String amka).* Η μέθοδος αυτή είναι υπεύθυνη να επιστρέψει σε μορφή μηνυμάτων, πιθανά λάθη που εντοπίστηκαν κατά την επικύρωση καταχωρήσεων **invalidLength()** και **invalidCharacters()**. Αν δεν εντοπιστεί λάθος επιστρέφει κενό String.
- *public static String **calculateAge**()*. Δηλώθηκαν οι τοπικές μεταβλητές **fifthDigit** και **sixthDigit**, τύπου integer που αντιστοιχούν στην θέση 4 και 5 του πίνακα **arrayAmkaDigits[]**. Για να πάρουμε τα ψηφία στις θέσεις αυτές και όχι τον κωδικό των χαρακτήρων σύμφωνα με το Unicode χρησιμοποιούμε την **.getNumericValue** της κλάσης **Character**. Επίσης δηλώθηκε η μεταβλητή **baseyear** την οποία χειρίστηκα με ternary operator, ζητώντας να συγκρίνει εάν το 5^ο ψηφίο του Α.Μ.Κ.Α είναι μεγαλύτερο ή ίσο του 2. Εάν ισχύει η συνθήκη, η **baseyear** θα γίνει 1900 ενώ εάν δεν ισχύει θα γίνει 2000. Τέλος η μεταβλητή **amkaOwnerYear** την οποία επιστρέφει η μέθοδος υπολογίζεται με τη δοθείσα μαθηματική πράξη.
- *public static String **calculateGender**()*. Δηλώθηκαν οι τοπικές μεταβλητές **genderDigit**, τύπου character η οποία αντιστοιχεί στη θέση 9 του πίνακα **arrayAmkaDigits[]**, και την μεταβλητή **gender** τύπου String, την οποία χειρίζομαι με ternary operator. Απομονώνοντας το 10^ο ψηφίο του Α.Μ.Κ.Α. και συγκρίνοντας εάν το υπόλοιπο της διαίρεσης του με το 2 ισούται με το μηδέν ή όχι, παίρνει τιμή **"Female"** ή **"Male"**
- *public static boolean **restartOrExit**()*. Η μέθοδος θα καλεστεί από την **main()** στο τέλος της **do_while**. Η μέθοδος δημιουργεί νέο αντικείμενο της κλάσης **Scanner** με όνομα **keyboard**. Εμφανίζει στο terminal το ερώτημα εάν επιθυμεί ο χρήστης να επαναλάβει την διαδικασία ή να τερματίσει το πρόγραμμα. Δημιουργείτε η μεταβλητή **answer** τύπου String η οποία ορίζεται ως η όποια επόμενη καταχώρηση στο terminal. Εάν είναι "Y" το πρόγραμμα θα τρέξει από την αρχή αλλιώς εάν πατηθεί οποιοδήποτε άλλο πλήκτρο θα τερματίσει το πρόγραμμα.
- *public static void **runAmkaProcess**()*. Δηλώθηκαν χωρίς αρχικοποίηση, δύο τοπικές μεταβλητές, την **boolean hasErrors** και την **String amka**. Το κομμάτι του κώδικα που εκτελείται μέσα στο block της **do_while**, με statement την μεταβλητή **hasErrors**, πρώτον ορίζει την μεταβλητή **amka** ως την καταχώρηση του χρήστη, που επιστρέφει η **interactWithUser()** μέθοδος και την μεταβλητή τύπου String **errors** ως το αλφαριθμητικό που επιστρέφει η **validateUserInput()**. Στη συνέχεια ορίζει την **hasErrors = errors.length() > 0** και η **do_while** loop εκτελείται έως ότου το αλφαριθμητικό που θα εισαχθεί από τον χρήστη να μην έχει λάθη κατά την καταχώρηση, δηλαδή όταν το **errors.length** ισούται με 0 (**errors.length==0**). Τέλος έγινε εισαγωγή δύο τοπικών μεταβλητών, της **age** τύπου integer η οποία παίρνει την τιμή που επιστρέφει η **calculateAge()** και της **gender** τύπου String η οποία παίρνει το String που επιστρέφει η **calculateGender()** και μετά ζητείται να εκτυπωθούν το έγκυρο Α.Μ.Κ.Α, η ηλικία του χρήστη και το φύλο του.

Θέμα 3: Παράθυρο Λογισμικού με χρήση περισσότερων της μίας κλάσεων

Να δημιουργήσετε πρόγραμμα Java που να περιέχει δύο κλάσεις Window και WindowApp. Η πρώτη αντιπροσωπεύει ένα παράθυρο λογισμικού και η δεύτερη είναι η κύρια κλάση η οποία περιλαμβάνει τη μέθοδο main. Ένα παράθυρο έχει όνομα (ένα αλφαριθμητικό), ύψος, πλάτος, χρώμα (ένα αλφαριθμητικό) και σημείο ορισμού (είναι το πάνω αριστερά σημείο του παραθύρου). Οι τιμές των συντεταγμένων είναι ακέραιες και θετικές ή μηδέν. Ως παράδειγμα ενός παραθύρου μπορούμε να θεωρήσουμε τις ακόλουθες τιμές: ύψος 12, πλάτος 8, λευκό χρώμα και σημείο ορισμού (0, 0).

Να ορίσετε την κλάση Window που θα περιλαμβάνει

- έναν κατασκευαστή για δημιουργία παραθύρων με μηδενικές ή κενές τιμές για τις μεταβλητές στιγμιότυπου (ανάλογα με τον τύπο κάθε μεταβλητής στιγμιότυπου), με εξαίρεση την τιμή του ονόματος του παραθύρου που θα του δίνεται ως παράμετρος,
- έναν κατασκευαστή για δημιουργία παραθύρων με τιμές για τις μεταβλητές στιγμιότυπου που θα του δίνονται ως παράμετροι,
- μια μέθοδο getName που θα επιστρέφει το όνομα του παραθύρου,
- μια μέθοδο getHeight που θα επιστρέφει το ύψος του παραθύρου,
- μια μέθοδο getWidth που θα επιστρέφει το πλάτος του παραθύρου,
- μια μέθοδο getColor που θα επιστρέφει το χρώμα του παραθύρου,
- μια μέθοδο resize που θα δέχεται ως ορίσματα 2 νέες τιμές και θα αλλάζει το ύψος και το πλάτος του παραθύρου,
- μια μέθοδο move που θα μετακινεί το σημείο ορισμού του παραθύρου σ' ένα άλλο σημείο που θα δίνεται ως όρισμα,
- μια μέθοδο isSquare που θα επιστρέφει τη διαγώνιο του παραθύρου αν το παράθυρο είναι τετράγωνο, αλλιώς -1,
- μια μέθοδο printWindow που θα τυπώνει την τρέχουσα κατάσταση του παραθύρου (όνομα, ύψος, πλάτος, χρώμα, σημείο ορισμού).

Να ορίσετε την κλάση WindowApp η οποία θα περιλαμβάνει τη μέθοδο main. Η main θα:

- δημιουργεί δύο στιγμιότυπα της Window, το πρώτο με όνομα "w1" και με μηδενικές ή κενές τιμές για τις υπόλοιπες μεταβλητές στιγμιότυπου, και το δεύτερο με τις τιμές: όνομα "w2", ύψος 15, πλάτος 10, χρώμα "blue" και σημείο ορισμού (2, 2),
- εκτυπώνει την τρέχουσα κατάσταση κάθε παραθύρου,
- μετακινεί το παράθυρο "w1" ώστε το σημείο ορισμού του να συμπίπτει με αυτό του "w2",
- εκτυπώνει την τρέχουσα κατάσταση του "w1",
- αλλάζει τα στοιχεία του "w2", ώστε να γίνει διπλάσιο στο πλάτος και μισό στο ύψος,
- εκτυπώνει την τρέχουσα κατάσταση του "w2".

➤ Υλοποίηση WindowApp.java

Το πρόγραμμα αποτελείται από δύο κλάσεις, την κλάση **Window** και την κλάση **WindowApp**. Έγινε import της java.awt.* για να χρησιμοποιήσω την κλάση Point, η οποία με βοήθησε στην διαχείριση των σημείων ορισμού των παραθύρων.

ΚΛΑΣΗ Window

Έγινε import της java.awt.* για να χρησιμοποιήσω την κλάση Point, η οποία με βοήθησε με τα σημεία ορισμού των παραθύρων.

A) Μεταβλητές Κλάσης Window

Οι 5 μεταβλητές στιγμιότυπου που δηλώθηκαν είναι οι εξής:

- *int height* -> ύψος παραθύρου λογισμικού
- *int width* -> πλάτος παραθύρου λογισμικού
- *String name* -> όνομα παραθύρου λογισμικού
- *String color* -> χρώμα παραθύρου λογισμικού
- *Point point* = *new Point (x:0, y:0)* -> σημείο ορισμού παραθύρου λογισμικού το οποίο αρχικοποιήθηκε με μηδενικές τιμές.

B) Ανάλυση Κατασκευαστών

Η κλάση αποτελείται από 2 κατασκευαστές, ως εξής:

- *protected Window (String name)*. Επιτελεί δημιουργία παραθύρων με μηδενικές ή κενές τιμές για τις μεταβλητές στιγμιότυπου, (ανάλογα με τον τύπο κάθε μεταβλητής στιγμιότυπου), με εξαίρεση την τιμή του ονόματος του παραθύρου που του δίνεται ως παράμετρος.
- *protected Window (String name, int height, int width, String color, int point_x, int point_y)*. Επιτελεί δημιουργία παραθύρων με τιμές για τις μεταβλητές στιγμιότυπου που του δίνονται ως παράμετροι.

Γ) Ανάλυση μεθόδων

Η κλάση αποτελείται 09 μεθόδους, ως εξής:

- *private String getName()*. Η μέθοδος επιστρέφει το όνομα του παραθύρου λογισμικού.
- *private int getHeight()*. Η μέθοδος επιστρέφει το ύψος του παραθύρου λογισμικού.
- *private int getWidth()*. Η μέθοδος επιστρέφει το πλάτος του παραθύρου λογισμικού.
- *private String getColor()*. Η μέθοδος επιστρέφει το χρώμα του παραθύρου λογισμικού.
- *private String getPoint()*. Η μέθοδος επιστρέφει το σημείο ορισμού του παραθύρου λογισμικού ως αλφαριθμητικό. Έγινε χρήση των μεθόδων getX() και getY() από το api της κλάσης Point για να επιτευχθεί η μορφή point = (x,y) κατά την εμφάνιση στο terminal.
- *protected void resize ()*. Η μέθοδος δέχεται ως ορίσματα 2 νέες τιμές και θα αλλάζει το ύψος και το πλάτος του παραθύρου
- *protected void move(int x, int y)*. Η μέθοδος μετακινεί το σημείο ορισμού του παραθύρου σ' ένα άλλο σημείο που θα δίνεται ως όρισμα, κάνοντας χρήση της .move της Point.
- *protected Double isSquare()*. Η μέθοδος επιστρέφει τη διαγώνιο του παραθύρου αν το παράθυρο είναι τετράγωνο σύμφωνα με την μαθηματική formula της μεθόδου, αλλιώς -1.
- *protected void printWindow()*. Εκτυπώνει την τρέχουσα κατάσταση κάθε παραθύρου.

ΚΛΑΣΗ WindowApp

Σε αυτήν την κλάση βρίσκεται η *public static void **main**(String[] args)*. Μέσα στη *main()* δημιουργούμε δύο παράθυρα **w1** και **w2** σύμφωνα με το ζητούμενο της άσκησης. Καλούμε την **printWindow()** για να εκτυπώσουμε την τρέχουσα κατάσταση των w1, w2. Καλούμε την μέθοδο **move()** για αλλαγή σημείου ορισμού w1->w2 και την μέθοδο **resize()** για αλλαγή διαστάσεων του w2. Εκτυπώνουμε την νέα κατάσταση των w1 και w2. Χωρίς να είναι ζητούμενο αποφάσισα να καλέσω και την **isSquare()** για να ελέγξω την λειτουργικότητα της.